CLAIMS:

1. An apparatus for implementing a Floating-Point related application, comprising:

a tool that includes:

a receiver for receiving a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

a parser for parsing the commands;

a processor configured to process at least the parsed commands for realizing the floating-point related application on the basis of said events.

2. The apparatus of Claim 1, wherein said language further defining regrouping of the events into at least one coverage model; and wherein said processor is configured to process the parsed commands for realizing the floating-point related application on the basis of said events and said at least one coverage model.

3. The apparatus according to Claim 1, wherein said application is an evaluation of coverage of tests being run on a design.

4. The apparatus according to Claim 1, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events.

5. The apparatus according to Claim 4, wherein said verification includes verifying if the Floating-Point operation complies with IEEE standard for Floating-Point.

6. For use with the Floating-Point module of Claim 1, a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction.

7. The computer language of Claim 6, further defining regrouping of the events into at least one coverage model.

8. An apparatus for implementing a Floating-Point related application, comprising:

a tool that includes:

a receiver for receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction;

a parser for parsing the commands;

a processor for processing at least the parsed commands for realizing the Floating -point related application at least on the basis of said events and said at least one coverage model.

9. The apparatus according to Claim 8, wherein said application is an evaluation of coverage of tests being run on a design.

10. The apparatus according to Claim 8, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events and the at least one coverage model.

11. The apparatus according to Claim 8, wherein said verification includes verifying if the Floating-Point operation complies with the IEEE standard for Floating-Point.

12. For use with the Floating-Point module of Claim 8, a computer language; the language defining\ Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction, the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction.

~~13.~~12. An apparatus for implementing a Floating-Point related application, comprising:

a tool that includes:

a receiver for receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

a parser for parsing the commands;

a processor for processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating -point related application.

14. The apparatus according to Claim 13, wherein said application is an evaluation of coverage of tests being run on a design.

15. The apparatus according to Claim 13, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events and at least one coverage model.

16. The apparatus according to Claim 13, wherein said verification includes verifying if the Floating-Point operation complies with the IEEE standard for Floating-Point.

17. For use with the Floating-Point module of Claim 13, a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on: (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers.

18. The apparatus according to Claim 1, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

19. The apparatus according to Claim 6, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

20. The apparatus according to Claim 7, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

21. The apparatus according to Claim 8, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number;

Relative Values of at least two FP numbers, and logical operations among said commands.

22. The apparatus according to Claim 12, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

23. The apparatus according to Claim 13, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

24. The apparatus according to Claim 17, wherein said list of commands includes: Range of FP numbers ; Mask on bits of FP number; Set or Reset Number of Bits in an FP number; Set or Reset Continuous-Bit-Long in an FP number; Relative Values of at least two FP numbers, and logical operations among said commands.

25. The apparatus according to Claim 8, wherein said constraints are further applied to attributes of Machine State.

26. The apparatus according to Claim 13, wherein said constraints are further applied to attributes of Machine State.

27. A method for implementing a Floating-Point related application that includes the steps of :

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing the floating-point related application on the basis of said events.

28. A method for implementing a Floating-Point related application that includes the steps of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction;

(ii) parsing the commands; and

(iii) processing at least the parsed commands realizing the Floating -point related application at least on the basis of said events and said at least one coverage model.

29. A method for implementing a Floating-Point related application, that includes the step of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating -point related application.

30. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application that includes the steps of :

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing the floating-point related application on the basis of said events.

31. A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

computer readable program code for causing the computer to parse the commands; and

computer readable program code for causing the computer to process at least the parsed commands for realizing the floating-point related application on the basis of said events.

32. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a

Floating-Point related application, that includes the steps of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating -point related application.

33. A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

computer readable program code for causing the computer to parse the commands; and

computer readable program code for causing the computer to process at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-point related application.

34. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application, that includes the steps of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating -point related application.

35. A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

computer readable program code for causing the computer to parse the commands; and

computer readable program code for causing the computer to process at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating -point related application.